Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computer and System Sciences 72 (2006) 1–15

JOURNAL OF COMPUTER AND SYSTEM SCIENCES

www.elsevier.com/locate/jcss

Real-valued multiple-instance learning with queries☆

Daniel R. Dooly[a], Sally A. Goldman[b,*,1], Stephen S. Kwek[c]

[a] Southern Illinois University Edwardsville, Edwardsville, IL 62026, USA
[b] Washington University, St. Louis, MO 63130, USA
[c] University of Texas San Antonio, San Antonio, TX 78249, USA

Received 22 March 2005; received in revised form 24 May 2005

Abstract

While there has been a significant amount of theoretical and empirical research on the multiple-instance learning model, most of this research is for concept learning. However, for the important application area of drug discovery, a real-valued classification is preferable. In this paper we initiate a theoretical study of real-valued multiple-instance learning. We prove that the problem of finding a target point consistent with a set of labeled multiple-instance examples (or bags) is NP-complete, and that the problem of learning from real-valued multiple-instance examples is as hard as learning DNF. Another contribution of our work is in defining and studying a multiple-instance membership query (MI-MQ). We give a positive result on exactly learning the target point for a multiple-instance problem in which the learner is provided with a MI-MQ oracle and a single adversarially selected bag.

Keywords: Learning theory; Multiple-instance learning; On-line learning; Membership queries

☆ Expanded version of a paper presented at the Twelfth International Conference on Algorithm Learning Theory (ALT 2001).

* Corresponding author.

E-mail addresses: ddooly@siue.edu (D.R. Dooly), sg@cse.wustl.edu (S.A. Goldman), kwek@cs.utsa.edu (S.S. Kwek)

URLs: http://www.siue.edu/~ddooly (D.R. Dooly), http://www.cse.wustl.edu/~sg (S.A. Goldman), http://www.cs.utsa.edu/~kwek (S.S. Kwek).

[1] This material is based upon work supported by the National Science Foundation under Grant No. IDM-0329241.

doi:10.1016/j.jcss.2005.06.002

1. Introduction

The *multiple-instance* (MI) learning model is becoming increasingly important within machine learning. Unlike standard supervised learning in which each instance is labeled in the training data, in the standard MI learning model each example is a set (or *bag*) of instances (or points) which is labeled as to whether any single point within the bag is positive. The individual points are not given a label. The goal of the learner is to generate a hypothesis to accurately predict the label of previously unseen bags.

The MI model was motivated by the *drug activity prediction problem* where each example corresponds to a molecule of interest and each bag contains all low-energy (and hence likely) configurations (or shapes) for the molecule [6]. There has been a significant amount of theoretical and empirical research directed towards this problem. Other applications for the multiple-instance model have been studied. For example, Maron and Ratan [13] applied the MI model to the task of learning to recognize a person from a series of images that are labeled positive if they contain the person and negative otherwise. They have also applied this model to learn descriptions of natural images (such as a waterfall) and then used the learned concept to retrieve similar images from a large image database. More recently, Ruffo [15] has used this model for data mining applications.

Most prior research performed under the MI model is for concept learning (i.e. Boolean labels). The first empirical study of Dietterich et al. [6] used real data for the problem of predicting whether or not a synthetic molecule binds to the musk receptor. However, binding affinity between molecules and receptors is quantitative, borne out in quantities such as the energy released by the molecule–receptor pair upon binding and hence a real-valued classification of binding strength in these situations is preferable. Dietterich et al. [6] say "The only aspect of the musk problem that is substantially different from typical pharmaceutical problems is that the musk strength is measured qualitatively by expert human judges, whereas drug activity binding is usually measured quantitatively through biochemical assays."

Furthermore, the previous work has just considered learning from a given set of labeled bags. However, in the real drug-discovery application, obtaining the label for a bag (which corresponds to making the drug and then running a laboratory experiment) is very time consuming. The process that would be used is to start with some "random" drug labeled as to whether or not it binds. Then a new drug is selected and created followed by an experiment to obtain its affinity value (i.e. the label), and so on. In the machine learning community, this learning protocol is called *active learning* since the learner gets to actively choose the next example for an expert (i.e. the experiment) to label selected from among a provided set of unlabeled examples. In learning theory research, active learning is modeled via a *membership query* which is an oracle that when given any unlabeled example from the domain will return the label for that example. Selecting the next drug to test is very much like a membership query (which outputs a real-valued label) except one cannot select an arbitrary set of points to define a bag but rather can only select a drug which in turn defines a bag.

Our goal here is to initiate a theoretical study on real-valued MI learning which includes the introduction of a MI membership query (MI-MQ). We prove that the problem of finding a target point consistent with a set of labeled MI examples (or bags) is NP-complete. We also prove that the problem of learning from real-valued MI data is as hard as learning DNF. A key contribution of this paper is a positive result on exactly learning the target point for a MI problem in which the learner is provided with a MI-MQ and a single adversarially selected bag $b = \{p_1, \dots, p_r\}$. The MI-MQ oracle takes as input any n-dimensional shift vector $\vec{v}$ and returns the distance label for $b + \vec{v} = \{p_1 + \vec{v}, \dots, p_r + \vec{v}\}$.

2. The real-valued multiple-instance model

Unlike standard supervised learning in which each instance is labeled in the training data, in this model each example is a set (or *bag*)[2] of instances which is labeled as to whether any single point within the bag is positive. The individual points are not given a label. The goal of the learner is to generate a hypothesis to accurately predict the label of previously unseen bags. Consider the standard learning problem of learning an axis-aligned box in $\Re^n$. In the standard learning model each labeled example is a point in $\Re^n$ (drawn according to some unknown distribution **D**) and labeled as positive if and only if it is in the target box. In the MI model, an example is a collection of points in $\Re^n$ (often called a *bag* or *r-example*) which is labeled as positive if and only if at least one of the points in the bag is in the target box. More formally, in the MI learning the training data $D = \{\langle B_1, \ell_1\rangle, \ldots, \langle B_m, \ell_m\rangle\}$ consists of a set of m bags where bag B_i has label ℓ_i. Let bag $B_i = \{B_{i1}, \ldots, B_{ij}, \ldots\ B_{in}\}$ where B_{ij} is the jth point in bag i. When all bags contain exactly r points then we use the terminology of *r-example*. Let ℓ_{ij} be the label for point B_{ij}. For the Boolean target concept of an axis-aligned box, ℓ_{ij} is 1 if and only if B_{ij} is in the target box. For the real-valued case, the target concept could be defined according to the distance between the center of the box and B_{ij}. The MI model assumes the label of the bag is determined by the point in the bag with the highest label. Hence, for Boolean labels, $\ell_i = \ell_{i1} \vee \ell_{i2} \vee \cdots \vee \ell_{in}$, and for real-value labels, $\ell_i = \max\{\ell_{i1}, \ell_{i2}, \ldots, \ell_{in}\}$.

The MI model was motivated by the *drug activity prediction problem* where each example is a possible configuration (or shape) for a molecule of interest and each bag contains all low-energy (and hence likely) configurations for the molecule [6,8]. For the drug-discovery application, each bag corresponds to a drug, each point in the bag corresponds to the shapes that it is likely to take, and the target point corresponds to the ideal shape that will create the strongest bond with the receptor molecule. By accurately predicting which molecules will bind to an unknown protein, one can accelerate the discovery process for new drugs, hence reducing cost.

We assume that there is a target point t in $\Re^n$ which corresponds to the ideal shape. A Boolean label then indicates whether or not the given conformation is close enough to t to bind. For target t and point p, let $dist(t, p)$ be the distance between t and p_i in the L_2 norm and V be a function that relates distance with binding strength. For example, V could be defined by the widely used empirical potential for intermolecular interactions, the Lennard-Jones potential $V(d) = 4\varepsilon\left(\left(\frac{\sigma}{d}\right)^{12} - \left(\frac{\sigma}{d}\right)^{6}\right)$ where ε is the depth of the potential well, σ is the distance at which $V(d) = 0$, and d is the internuclear distance for two monoatomic molecules [4]. The Lennard-Jones model is nice because of its mathematical simplicity and its ability to qualitatively mimic the real interaction between molecules. For the purposes of this paper, the only property we assume about the computation of the binding strength between p and q is that from it $dist(p, q)$ can be computed and that the binding strength diminishes as the distance to the target increases. Then the label for bag $b = \{p_1, \ldots, p_r\}$ is $\max_{i=1,\ldots,r} V(dist(t, p_i))$ An alternate definition for the label of $b = \{p_1, \ldots, p_r\}$ is to compute

$$d_{\min}(b) = \min_{i=1,\ldots,r} dist(t, p_i)$$

[2] We use the standard terminology of the field, in which a bag of points is a set, not a mathematical bag, of points.

and then return $V(d_{\min}(b))$ as the label. We will use this view and further, assume that $d_{\min}$ itself is given. In general, one can extend this model by using a weighted L_2 norm but in this work we assume that an unweighted L_2 norm is used.

While most MI learning algorithms assume Boolean labels, recent work studies extensions of the DD and citation k-NN algorithms for data with real-value labels [1] and MI regression [14]. The MI regression work assumes an underlying linear model for the hypothesis and thus has a different inductive bias. Zhang and Goldman developed EM-DD which combines the DD algorithm with the expectation-maximization (EM) algorithm [18]. More recently, Andrews et al. [2] combined EM with a support vector machine to develop MI learning algorithms.

The most natural generalization of the standard single-instance membership query for the MI model would be to allow as input to the MI membership oracle an arbitrary bag (perhaps of a some fixed size r). There are several reasons why this is not a good way to model the MI membership query. First, if allowed to do this then by perturbing the individual points in a given bag b, the learning algorithm could determine which point is closest to the target which would effectively reduce the problem to a single-instance problem. Secondly, as discussed earlier, in reality one can select a drug (which could be a small variation of an earlier drug tested). However, the set of bags that correspond to real drugs are limited and in general there will not exist a drug that would have as its likely conformations an arbitrary r points. In particular, a molecule smoothly moves between conformations (shapes) and thus there is some dependency among the points in the bag. However, the dependency is very complex and thus defining a MI membership query that captures the physical constraints of the underlying chemistry is challenging and we do not claim to have solved that problem here.

We now define our MI membership oracle (MI-MQ). Given a bag $b = \{p_1, \ldots, p_r\}$ where b is provided by an adversary, we define the MI-MQ oracle to be one that takes as input any n-dimensional shift vector $\vec{v}$ and returns the real-valued label for $b + \vec{v} = \{p_1 + \vec{v}, \ldots, p_r + \vec{v}\}$. While this model does not capture all of the physical constraints of the underlying chemistry, it does maintain the relationship between the points in the provided bag b (which is adversarially provided and thus could always be one obtained from a real drug) since the same vector is added to every point in b. We feel that this proposed model is a good starting point for developing a theory of learning with queries for real-valued MI learning. Developing a model that better captures the physical constraints of the underlying chemistry and thus would enable the needed laboratory experiments required of the MI-MQ to be performed in a cost-effective way is an intersting direction for future work.

3. Prior work

We begin with a summary of the prior work on learning the (Boolean) MI concept class of axis-aligned boxes in n-dimensional space. To understand the distributional assumptions made by the prior work, we begin with some definitions. We use D to denote an arbitrary distribution over n-dimensional points. We use $\mathbf{D}$ to denote a distribution over r-examples. For any point $\vec{p}$ and distribution $\mathbf{D}$ over r-examples, let $w_{\mathbf{D}}^{+}(\vec{p})$ denote the conditional probability that $\vec{p}$ is a point in a random bag drawn according to $\mathbf{D}$ given that bag has a positive label. For any region $R \subseteq \Re^n$, let $w^{+}(R) = \int_{\vec{p} \in R} w_{\mathbf{D}}^{+}(\vec{p})$. Long and Tan [10] described an efficient PAC algorithm under the restriction that each point in the bag is drawn independently from a product distribution, D_{product}. Hence the resulting distribution over r-examples is $\mathbf{D} = D_{\text{product}}^{r}$. Auer et al. [3] gave an efficient PAC algorithm that allows each point to be drawn independently from an arbitrary distribution D. Hence each r-example is drawn from $\mathbf{D} = D^r$ for an arbitrary distribution D defined over points. In their paper, Auer et al. [3] also proved that if the distributional requirements are further relaxed to allow an arbitrary distribution $\mathbf{D}$ over r-examples then learning axis-aligned boxes is as hard as learning DNF formulas in the PAC model. Blum and Kalai [5] described a simple reduction from the problem of PAC learning from MI examples to that of PAC learning with one-sided random classification noise when the r-examples are drawn from D^r for D any distribution defined over points. They also described a more efficient (and more involved) reduction to the statistical-query model [9] that yields the most efficient PAC algorithm known for learning axis-aligned boxes in the MI model over D^r for an arbitrary distribution D defined over points. Their algorithm has sample complexity $\tilde{O}(n^2 r/\varepsilon^2)$, roughly a factor of r faster than the result of Auer et al.

To understand some of the difficulties that occur when switching from the Boolean to real-valued setting, we briefly overview the basic technique used to obtain these results. Let t be the target box, and define $T = \{\vec{p} \mid \vec{p} \in t\}$. The key property used by earlier results is that there is some constant c such that for any region R where $R \cap T = \emptyset$, $w^+_{\mathbf{D}}(R) = c$. For all known positive results, $\mathbf{D} = D^r$. Under this definition of $\mathbf{D}$, the needed property holds since any point in a positive bag that is in region R where $R \cap T = \emptyset$ must be in a bag with some point in T. In other words, one of the other $r - 1$ points in the bag (drawn independently from the same distribution) must have been in T. It is under this iid assumption that Blum and Kalai [5] show how to reduce MI learning to PAC learning with one-sided noise.

We now consider the real-valued setting where the label for bag b is a function of the distance between the closest point in the bag b and the target. In this setting, the sharp change that occurs in the fraction of positive examples as a half-space crosses the boundary of the box (in the Boolean domain) is no longer present. Hence, a completely different approach appears to be needed. However, in order to ensure that we obtain an algorithm that is polynomial for an arbitrary number of dimensions, we must in some way be able to independently work with each dimension (or at least a constant number of dimensions at a time).

The only theoretical work which we are aware of that studies real-valued MI learning is work by Goldman and Scott [7]. Similar to our work here, they associate a real-valued label with each point in the MI example. These values are then combined using a real-valued aggregation operator to obtain the classification for the example. Here, we only consider the minimum for the aggregation operator. They provide on-line agnostic algorithms for learning real-valued MI geometric concepts defined by axis-aligned boxes in *constant* dimensional space by reducing the learning problem to one in which the exponentiated gradient (or gradient descent) algorithm can be used. However, their work (and their basic technique) assumes that d is constant which is not feasible for the drug discovery application since d is typically in the hundreds.

Most empirical work also considers the Boolean setting. In their seminal paper, Dietterich et al. [6], presented three methods for learning axis-aligned boxes (often referred to as APR for axis-parallel rectangles) in the MI model. They presented an algorithm they refer to as the "outside-in" algorithm. In this algorithm, first they construct the smallest box that bounds all of the positive examples, and then shrinks this box to exclude false positives. Finally, they presented a third algorithm, the "inside-out" algorithm which starts with a point in the feature space and "grows" a box with the goal of finding the smallest box that covers at least one example from each positive bag and no examples from any negative bag. Then they expand the resulting box (via a statistical technique) to get better results. When appropriately tuned, their algorithm gives 89% accuracy on the Musk2 data set.

The work of Dietterich et al. [6] was preceded by the work of Jain et al. [8] in which they presented COMPASS which is an APR-like neural network algorithm which is robust to errors in the initial alignment of the molecules. While COMPASS can handle real-valued labels, we are not aware of any reported results on any available real-valued data sets.

Auer et al. [3] presented an algorithm that learns using simple statistics and hence avoids some potentially hard computational problems that were required by the heuristics used by Dietterich et al. Their algorithm worked quite well on the Musk2 data set (obtaining a 84% accuracy) despite the fact that they assumed each point in a bag was drawn independently of the others.

Maron and Lozano-Pérez [12] described a framework called *Diverse Density* (see also Maron [11]). The intuition of their approach is as follows. When describing the shape of a molecule by n features, one can view each configuration of the molecule as a point in a n-dimensional feature space. As the molecule changes its shape, it traces out a manifold through this n-dimensional space. (To keep the size of the bags manageable, only shapes of the molecule that have sufficiently low potential energy were considered.) The diverse density at a point p in the feature space is a measure of both how many *different* positive bags have an example near p, and how far the points from negative bags are from p. They use gradient ascent with multiple starting points (namely, starting from each point from a positive bag) to find the point that maximizes the diverse density. Their algorithm obtained 82.5% accuracy on the Musk2 data.

Amar et al. [1] and Zhang and Goldman [18] empirically studied diverse-density-based and k-citation nearest neighbor-based algorithms for learning in the real-valued MI model. However, even for the original versions of the diverse density [12] and k-citation nearest neighbor algorithms [16] for the Boolean domain, no theoretical results have been shown.

Wang and Zucker [16] proposed a lazy learning approach to MI learning by applying a variant of the k nearest neighbor algorithm (k-NN). To compute the distance between bags b_1 and b_2 they used the minimum distance between a point in b_1 and a point in b_2. While a standard k-NN approach did not work well, by also using citers of p (points who include p as one of its nearest-neighbors) as well as p's nearest neighbors they reached a 92.4% accuracy on Musk1 and 86.3% accuracy on Musk2.

Ray and Page [14] studied MI linear regression using artificial data to empirically evaluate their algorithm which uses an inductive logic programming-based approach combined with a linear regression algorithm supplemented with expectation maximization. More recently, Warmuth et al. [17] have used support vector machines with active learning and real data. Again, no theoretical results are given in their work. The goal of our work here is to begin developing theoretical foundations for the real-valued MI model for high-dimensional spaces.

4. Results for the real-valued multiple-instance model

For the reminder of this paper we study the real-valued MI problem where we assume that each bag is drawn from an arbitrary distribution **D** and can have any number of examples within it. We define the *Real-Valued Multiple-Instance L2-Consistency Problem* as the following problem. As input you are given a set S of bags each labeled with a real value. The problem is to determine whether or not there is some target point $t \in \Re^n$ such that the label given to each bag is consistent with target t where we assume bag $b = \{p_1, \ldots, p_r\}$ for target t would receive the label $\min_{i=1,\ldots,r} \mathit{dist}(t, p_i)$ with the L_2 norm for the distance metric.

4.1. Negative results

In this section we present some negative results demonstrating the general MI learning problem is hard.

Theorem 1. *The real-valued MI L2-consistency problem is NP-complete.*

Proof. The proof is by reduction from 3-Sat. The instance space has n dimensions, one for each variable. The 3-Sat formula is transformed into a collection of bags as follows. For each clause in the formula, we introduce a bag of 3 points and assign it a label corresponding to a distance of $\sqrt{n-1}$. Each of these points corresponds to a literal in the clause with all coordinates set to 0 except for the coordinate that corresponds to the literal. If the corresponding literal is a negated literal $\overline{x}_i$ then the ith coordinate is set to -1, otherwise it is set to 1. In addition, we also add to this collection a bag $O = \langle 0, \ldots, 0\rangle$ with a distance label of $\sqrt{n}$.

Suppose the point $p = (p_1, \ldots, p_n)$ labels these bags consistently. Let $s = (s_1, \ldots, s_n)$ be the closest point to t an arbitrary point in a bag corresponding to a clause such that $dist(p, s) = \sqrt{n-1}$. Recall that every such bag holds three points and for each point all features are 0 except for one which is either set to 1 or -1. Let i be the nonzero feature of s. Then,

$$dist(p, s) = \sqrt{(s_i - p_i)^2 + \sum_{j \neq i} p_j^2} = \sqrt{(s_i - p_i)^2 - p_i^2 + \sum_{j=1}^{n} p_j^2} = \sqrt{n-1}.$$

Since p is consistent with the distance labels on the collection of bags,

$$dist(p, O) = \sqrt{\sum_{j=1}^{n} p_j^2} = \sqrt{n}.$$

Putting these two facts together yields $(s_i - p_i)^2 - p_i^2 + n = n - 1$ which simplifies to $s_i^2 - 2s_i p_i = -1$. Recall that by our construction either $s_i = 1$ or $s_i = -1$. It is easily verified that when $s_i = 1$ is substituted into the above equation, one obtains that $p_i = 1$. Likewise, when $s_i = -1$ is substituted into the above equation, one obtains that $p_i = -1$. Hence it follows that $p_i = s_i$.

Therefore, if there is a point which labels the bags consistently, we transform it into an assignment of variables which satisfies all the clauses as follows: if the coordinate of the point in dimension i is -1, assign false to variable x_i. If the coordinate of the point in dimension i is 1, assign true to variable x_i. Otherwise assign either true or false, at random. For each clause, at least one of the three relevant coordinates of the point will cause an assignment to a variable which makes that clause true. So the assignment satisfies all the clauses.

If there is an assignment of variables which satisfies all the clauses, then the point with coordinate 1 in dimensions corresponding to true variables and coordinate -1 in dimensions corresponding to false variables will meet all the distance criteria, since it is at distance $\sqrt{n}$ from the origin, and there will be at least one of the three points in each bag for which it is at distance $\sqrt{n-1}$. □

Theorem 1 does not indicate that learning is hard, but only that any learning algorithm that requires the consistency problem to be solved is not feasible. We now give a hardness result showing that the real-valued MI learning problem is as hard as learning DNF even if the learner is allowed to use a hypothesis

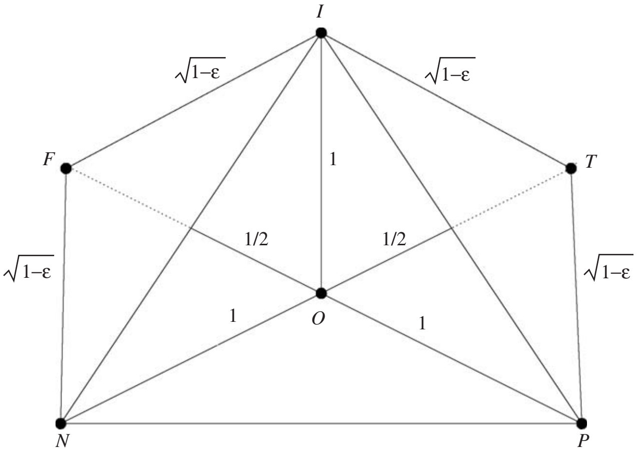


Fig. 1. The geometry of points I, P, N, F, T, O for the proof of Theorem 2.

class that is not simply a point in $\Re^n$. The statement of this result is very similar to the hardness result for the Boolean MI model of Auer et al. [3]. We note that our result does not follow from their results since each bag in the Boolean model is labeled as positive if and only if it is in the target box. Here, each bag must be labeled with the distance between its closest point and the target point. Hence, neither result subsumes the other.

Theorem 2. *Given a MI sample S for an unknown target box in $\Re^n$ where each $s \in S$ has a real-value label of the L2-distance between the target t and the closest point in s to t, the task of finding any polynomial evaluatable hypothesis that would correctly classify all points in S is as hard as learning DNF.*

Proof. This proof is by reduction from the problem of learning r-term DNF to the problem of learning in the real-valued MI setting. For ease of exposition, we assume that $n > 2$ and use $v_1, \ldots, v_n$ to denote the variables.

First, we describe how the reduction works for $r = 1$. Each literal in the term is represented by a point in the two-dimensional Euclidean plane with origin O. We use the following widget. Let points P (for positive), N (for negative) and I (for irrelevant), respectively, be the vertices of an equilateral triangle in the two-dimensional Euclidean plane that is centered around O where $I = (0, 1)$ and the distance from O to each of P, N, an I is 1 (see Fig. 1). Let $\varepsilon = \frac{1}{2n}$. Let T (for true) be a point that lies outside of $\triangle INP$ along the bisector of the segment $\overline{IP}$ that is $\sqrt{1-\varepsilon}$ away from both P and I. Similarly, let F (for false) be a point that lies outside the triangle and is $\sqrt{1-\varepsilon}$ away from both N and I. Observe that since $n > 2$, $\varepsilon < \frac{1}{4}$ which guarantees that T and F lie strictly outside of the $\triangle INP$.

The function g maps a term ϕ into a $2n$-dimensional point $g(\phi) = p = \langle p_1, p_2, \ldots, p_n\rangle$ where p_i is a two-dimensional point defined as follows. If v_i is in ϕ then $p_i = P$, if $\overline{v_i}$ is in ϕ then $p_i = N$, and finally, if v_i is not in ϕ then $p_i = I$. Similarly, for an example $x = (x_1, \ldots, x_n)$ is mapped to a $2n$ dimensional point by $g(x) = \langle g(x_1), \ldots, g(x_n)\rangle$ where $g(x_i) = T$ if $x_i = 1$ and $g(x_i) = F$ if $x_i = 0$.

Suppose that x satisfies ϕ then it is easily verified that $dist(g(\phi), g(x)) = \sqrt{n(1-\varepsilon)} = \sqrt{n - 1/2}$ since by construction the distance between each of the pair of points corresponding to the ith literal and

ith element of x is $\sqrt{1-\varepsilon}$. However, if x does not satisfy ϕ then in at least one two-dimensional space the distance between that portion of $g(\phi)$ and $g(x)$ has distance greater than $\frac{3}{2}$ and hence $dist(g(\phi), g(x)) > \sqrt{\frac{9}{4}+(n-1)(1-\varepsilon)} > \sqrt{n}$. To change the latter inequality to an equality so that all the negative examples in the DNF learning problem are mapped into points with unique distance values, we treat $g(x)$ as a bag and add the origin O as an additional point in $g(x)$. Clearly, $dist(g(\phi), O) = \sqrt{n}$ for all possible choices of ϕ and hence $dist(g(\phi), g(x)) = \sqrt{n}$ if x does not satisfy ϕ.

We now extend the reduction to r-term DNF $\phi = \phi_1 \vee \cdots \vee \phi_r$. In our transformation we introduce a two-dimensional subspace for each of the rn pairing of a variable with a term giving $2rn$ dimensions in all. For $1 \leqslant i \leqslant n$ and $1 \leqslant j \leqslant r$, we let S_{ij} to denote the subspace associated with the ith variable of term j. We transform the given r-term DNF formula ϕ into a target point in $2rn$-dimensional space as follows. If the ϕ_j contains a positive variable x_i, we set S_{ij} to P. If ϕ_j term contains $\overline{x}_i$ we set S_{ij} to N. Finally, if variable x_i does not appear in ϕ_j, we set S_{ij} to I. Thus, $g(\phi) = \langle S_{11} \cdots S_{1r} \;\; S_{21} \cdots S_{2r} \;\; \cdots \;\; S_{n1} \cdots S_{nr} \rangle$. Assignment $x = (x_1, \ldots, x_n)$ is mapped into a bag of $B(x)$ of $r+1$ points. This bag contains the origin O as well as points p_k for $1 \leqslant k \leqslant r$. Point p_k is defined as follows. For $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant r$, in p_k subspace S_{ij} is set to O if $j \neq k$, and for $j = k$, S_{ij} is set to T if $x_i = 1$ and set to F if $x_i = 0$.

We consider an example with $n = 4$ and $r = 2$. Let $\phi = (x_1 \wedge \overline{x}_2 \wedge x_4) \vee (x_1 \wedge x_3)$. Then $g(\phi) = PNIP\ PIPI$ where the first four pairs correspond to the first term and the second four pairs correspond to the second term. The assignment $x = 1001$ would be translated to the bag

$$g(x) = \{TFFT\ OOOO, OOOO\ TFFT, OOOO\ OOOO\}.$$

Each of the first r points in $g(x)$ tests to see if x satisfies term j, while the last is a reference point known to be closer to the target than the point corresponding to any unsatisfied term. To complete the transformation, we give the positive bags the distance value $\sqrt{rn - n\varepsilon} = \sqrt{rn - 1/2}$ and the negative bags the distance value $\sqrt{rn}$. We also include rn bags containing three points each. In each group of three, the coordinates in one of the subspaces S_{ij} are assigned to P, N, or I, and the coordinates in all other subspaces are assigned to O. Each of these rn bags has value $\sqrt{rn-1}$. Finally, we have a bag containing the point O with a distance value $\sqrt{rn}$.

As for the case of conjunctions it can be easily verified that if x satisfies ϕ, then $dist(g(x), g(\phi)) = \sqrt{n(1-\varepsilon)+(r-1)n} = \sqrt{rn - n\varepsilon} = \sqrt{rn - \frac{1}{2}}$. Conversely, if x falsifies ϕ_i then

$$dist(g(x), g(\phi)) \geqslant \sqrt{(n-1)(1-\varepsilon) + 9/4 + (r-1)n} > \sqrt{rn}.$$

Since $dist(O, g(\phi)) = \sqrt{rn}$ it follows that

$$dist(g(x), g(\phi)) = \begin{cases} \sqrt{rn - 1/2}, & x \text{ satisfies } \phi, \\ \sqrt{rn}, & x \text{ does not satisfy } \phi \end{cases}$$

as desired.

Suppose that we have an algorithm which is able to find a $2rn$-dimensional point p that has a distance to each provided bag where the distance equals the specified distance label. For any $2rn$-dimensional point p, we use p_{ij} to denote the value of p for subspace S_{ij}. Let q_{ij} be the one of P, N or I which is closest to p_{ij}. We now argue that the distance between p_{ij} and q_{ij} is zero. That is, p_{ij} must be one of P, N or I. From the bag with O for all subspaces, we have $\sum_{i,j} dist(p_{ij}, O)^2 = rn$. Multiplying both sides by $rn-1$ yields

$$(rn-1)\sum_{i,j} dist(p_{ij},O)^2 = rn(rn-1). \tag{1}$$

From the *rn* bags with three points each we have that

$$dist(p_{k\ell},q_{k\ell})^2 + \sum_{i\neq k, j\neq \ell} dist(p_{ij},O)^2 = rn-1.$$

Summing over the *rn* subspaces we get

$$\sum_{k\ell} dist(p_{k\ell},q_{k\ell})^2 + \sum_{k\ell}\sum_{i\neq k, j\neq \ell} dist(p_{ij},O)^2 = rn(rn-1).$$

Using the observation that

$$\sum_{k\ell}\sum_{i\neq k, j\neq \ell} dist(p_{ij},O)^2 = (rn-1)\sum_{ij} dist(p_{ij},O)^2$$

gives

$$\sum_{k\ell} dist(p_{k\ell},q_{k\ell})^2 + (rn-1)\sum_{ij} dist(p_{ij},O)^2 = rn(rn-1). \tag{2}$$

Combining Eqs. (1) and (2) gives that $\sum_{k\ell} dist(p_{k\ell},q_{k\ell})^2 = 0$ and hence $p_{k\ell}$ must be one of *P*, *N* or *I*.

Let us now consider a positive bag (i.e. a bag with label $\sqrt{rn-n\varepsilon}$). One of the points in this bag must be at distance $\sqrt{rn-n\varepsilon}$ from the target point $t = g(\phi)$. Let it correspond to term *j* and let us call it *z*. Since we know that $dist(z_{ik},O) = 1$, we can subtract the distance in all the subspaces except those corresponding to term *j* to get $\sum_i dist(t_{ij},z_{ij})^2 = n(1-\varepsilon)$. So each variable *i* must satisfy term *j*. Let us consider a negative bag. All of the points in this bag must be at least distance $\sqrt{rn}$ from *t*. Let us pick a point *w* corresponding to term *j*. There must be at least one subspace for which $dist(t_{ij},w_{ij})^2 > 1$. The only way this can happen is for variable *i* to fail to satisfy term *j*. So we can read the terms of the DNF from the values that *t* takes in the subspaces. If t_{ij} has location *P*, then term *j* contains literal x_i. If t_{ij} has location *N*, then term *j* contains the literal $\overline{x}_i$. Finally, if t_{ij} has location *I*, then term *j* does not contain include x_i (or its negation). □

4.2. Our positive result

In this section we present a positive result. Let *b* be an arbitrary bag provided by an adversary. We assume that we have access to a MI-MQ oracle and that from the label provided by this oracle we can then compute the distance between the closest point in $b+\vec{v}$ and the target *t* where $\vec{v}$ is the input given to the MI-MQ. It is important to remember that although we can compute the distance between the target and the closest point from $b+\vec{v}$ this provides no information as to which point in $b+\vec{v}$ is closest to the target.

A natural approach to use to solve this problem is the following. Suppose there was a single closest point *p* in bag *b* to the target, and further that we knew a value of ε so that the distance between *p* and the

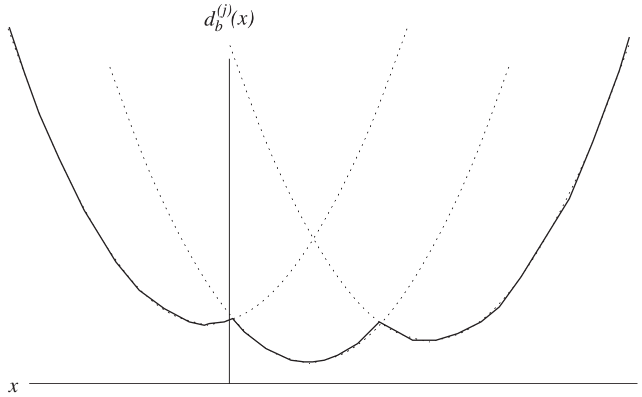


Fig. 2. A visualization of $d_b^{(j)}(x)$.

target is smaller by an additive factor of ε than the distance between the target and any other point in b. Then one can try shifting b in each direction by ε using membership query to find some shift for which p is closer to the target. By repeating this process one can bring p arbitrarily close to the target. Finally, once a shift is found so that p has distance 0 to the target, it is possible to actually determine the coordinate of the target. While this approach sounds simple, there are many important details that must be worked out. For example, to ensure that there is a single point p in b closest to the target it is necessary to repeat the process with small random shifts and thus the resulting algorithm becomes a randomized algorithm. An appropriate small value for ε must be computed and also, it is nontrivial to find the target point even when a bag with distance 0 from the target is found. So while it is possible to fill in the many missing details to the above approach, the algorithm that we give is much more efficient and is also deterministic.

We now describe our algorithm. The high-level approach is to independently determine the coordinate in each dimension of the target point. In order to describe our algorithm in more depth, we introduce the following definitions[3]. Let $p = (p_1, \dots, p_r)$ be an arbitrary point and let $\vec{v_j}$ be a unit vector along dimension j. For x an arbitrary real, we define $d_p^{(j)}(x)$ as the distance between the target t and the point $p + x \cdot \vec{v_j} = (p_1, \dots, p_{j-1}, p_j + x, p_{j+1}, \dots, p_r)$. That is,

$$d_p^{(j)}(x) = \sum_{i \neq j} (t_i - p_i)^2 + (t_j - (p_j + x))^2 = y_p^{(j)} + (x - m_p^{(j)})^2 \tag{3}$$

for constants (with respect to x) of $y_p^{(j)} = \sum_{i \neq j} (t_i - p_i)^2$ and $m_p^{(j)} = (t_j - p_j)$. Finally, let $d_b^{(j)}(x) = \min_{p \in b} d_p^{(j)}(x)$ and $p_b^{(j)}(x) = \operatorname{argmin}_{p \in b} d_p^{(j)}(x)$. That is, $d_b^{(j)}(x)$ is obtained by combining the r parabolas given by $d_p^{(j)}(x)$ for points $p_1, \dots, p_r$ in b. For each value of x, the value of $d_b^{(j)}(x)$ is that of the parabola that has the minimum value when evaluated at the dimension j shift of x. In other words, for $p = p_b^{(j)}(x)$, the value of $d_b^{(j)}(x)$ is $y_p^{(j)}$. Fig. 2 shows a visualization of $d_b^{(j)}(x)$.

[3] These definitions depend on the target t and bag b. However, for ease of exposition we do not explicitly include t and b in the notation.

Suppose for a moment we could find three offsets x_1, x_2, x_3 for which the same point p defines the minimum value of $d_b^{(j)}(x)$. In the next lemma, we argue that with knowledge of p, we could then find the dimension j value, t_j, for the target point t.

Lemma 3. *Let x_1, x_2 and x_3 be three translations and let p be a known point in b such that $p_b^{(j)}(x_1) = p_b^{(j)}(x_2) = p_b^{(j)}(x_3) = p$. Then the coordinate of the target in dimension j can be computed from the values returned by* MI-MQ$(b + x_1 \cdot \vec{v}_j)$, MI-MQ$(b + x_2 \cdot \vec{v}_j)$ *and* MI-MQ$(b + x_3 \cdot \vec{v}_j)$

Proof. By the fact that $p_b^{(j)}(x_1) = p_b^{(j)}(x_2) = p_b^{(j)}(x_3) = p$ it follows that for all three translations x_1, x_2 and x_3, p is the closest point to the target. For $i = 1, 2, 3$, let y_i be the value returned by MI-MQ$(b+x_i \cdot \vec{v}_j)$. Then $(p + x_1, y_1)$, $(p + x_2, y_2)$ and $(p + x_3, y_3)$ are all points on the parabola that takes on the minimum value of $y_p^{(j)}$ and the target value in dimension j is $t_j = p_j + m_p^{(j)}$. □

Our goal is now to independently, for each dimension j, find three translations that satisfy the conditions of Lemma 3 from which we can compute the target $t = (t_1, \ldots, t_n)$. As discussed above, in $d_b^{(j)}(x)$ there are r parabolas, one for each point in b. For each parabola it reaches a minimum value for the value of x that represents the dimension j shift for which $t_j - p_j = 0$. It is important to note (as shown above) that all r parabolas are of the form $y + (x - m)^2$ where y and m may be different for each of the points. In particular, for point $p \in b$, $y_p^{(j)}$ is the label for bag b that would be obtained if bag b were shifted in dimension j so that $t_j - p_j = 0$ and $m_p^{(j)}$ is the value of x where this parabola reaches its minimum value. Our next lemma shows that as we translate far enough in dimension j so that the closest point to the target will be the one with the minimum coordinate in dimension j.

Lemma 4. *Let $\vec{v}_j$ be a unit vector along dimension j. For bag b and dimension j, let x_m be the smallest value in dimension j among all points in b. Let $L = \{p \in b \mid p_j = x_m\}$. Let $x_r > x_m$ be the second smallest value in dimension j among all points in b. We define the target distance $d_\ell = \min_{p \in b} dist(p, t)$ for t the target point, $d_\Delta = x_r - x_m$, and the diameter $d_d = \max_{p_1, p_2 \in b} dist(p_1, p_2)$.*

For any $x > \frac{(d_\ell + d_d)^2 - d_\Delta^2}{2d_\Delta} + d_\ell + d_d$, the closest point in $b + x\vec{v}_j$ to the target is a point in L.

Proof. The distance between any point in $b + x\vec{v}_j$ and the target can be expressed in terms of the component along $\vec{v}_j$ and the component normal to $\vec{v}_j$ (See Fig. 3). Let d_p be defined as the maximal projection of the distance along the hyperplane, normal to $\vec{v}_j$, between any point in L and any point $p \in b - L$. We denote the component along $\vec{v}_j$ from any point $p \in L + x\vec{v}_j$ to the target as

$$d_v = x + (p - t) \cdot \vec{v}_j > \frac{(d_\ell + d_p)^2 - d_\Delta^2}{2d_\Delta} + d_\ell + d_d + (p - t) \cdot \vec{v}_j.$$

Since $p \in b$ it follows that the distance between the target t and p is at most $d_\ell + d_d$. Thus $(p - t) \cdot \vec{v}_j \geqslant -(d_\ell + d_d)$. Thus from this inequality, the fact that $d_d \geqslant d_p$,and the above inequality it follows that

$$d_v = \frac{(d_\ell + d_d)^2 - d_\Delta^2}{2d_\Delta} + \Delta$$

for some $\Delta > 0$.

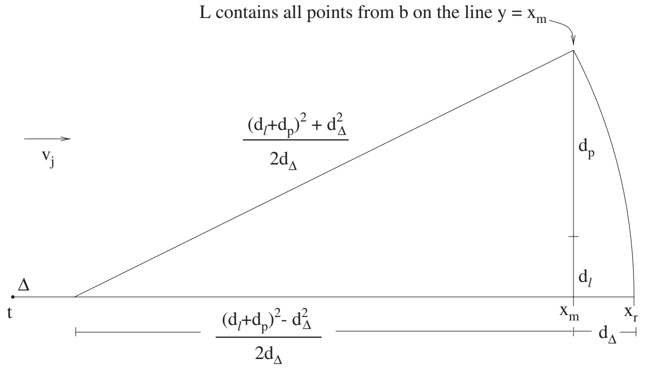

L contains all points from b on the line y = x_m
$\frac{(d_l+d_p)^2 + d_\Delta^2}{2d_\Delta}$
v_j
d_p
d_l
Δ
t
x_m
x_r
$\frac{(d_l+d_p)^2 - d_\Delta^2}{2d_\Delta}$
d_Δ

Fig. 3. The geometry for the proof of Lemma 4. All points in L have a jth coordinate of x_m which is the smallest value of the jth coordinate among all points in bag b. All points $p \in b \cdot L$ have a jth coordinate of at least x_r.

The distance from any point $p \in (b - L)$ to t is at least that of its component along $\vec{v_j}$, which is at least $d_v + d_\Delta$. We now consider only the points in L. For any point $p \in L$, the distance between p and t is at most

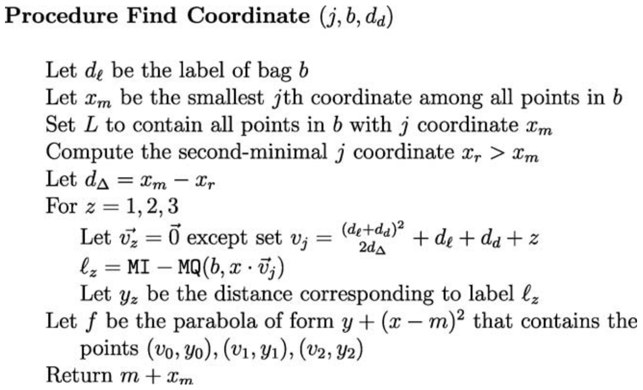
$$
\begin{aligned}
\sqrt{d_v^2 + (d_\ell + d_p)^2} &\leqslant \Delta + \sqrt{\left(\frac{(d_\ell + d_p)^2 - d_\Delta^2}{2d_\Delta}\right)^2 + (d_\ell + d_p)^2} \\
&= \Delta + \sqrt{\frac{(d_0 + d_2)^4}{4d_1^2} - \frac{(d_0 + d_2)^2}{2} + \frac{d_\Delta^2}{4} + (d_\ell + d_p)^2} = \Delta + \frac{(d_\ell + d_p)^2 + d_\Delta^2}{2d_\Delta} \\
&= \Delta + \frac{(d_\ell + d_p)^2 - d_\Delta^2}{2d_\Delta} \\
&\quad + d_\Delta < d_v + d_\Delta. \qquad \square
\end{aligned}
$$

We now describe how we use the above lemmas in the procedure Find_Coordinate (see the detailed pseudo-code in Fig. 4) to find the target value t_j for dimension j. First $d_\ell, d_\Delta, d_d, x_m, x_r$ and L are computed from the training data. We then use the MI-MQ oracle to query the value of $s_{\vec{t},b}^{(j)}(x)$ for three points, farther out. These points will lie on a parabola. The minimal value of the parabola gives us the j-coordinate of the target.

We now consider the procedure Find_Target which is the overall procedure to learn the target point (see Fig. 5). It independently finds the coordinates of the target in each of the n dimensions.

Theorem 5. *Assuming that each call to the MI-MQ oracle takes constant time, Find_Target has a worst-case time complexity of $O(nr^2)$ and is guaranteed to output the target point t where r is the number of points in the provided bag b and n is the number of dimensions for each point in b.*

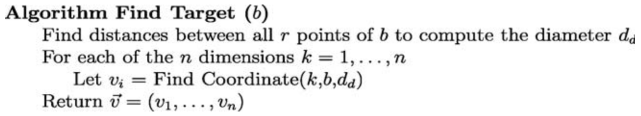
Proof. In Find_Coordinate, computing the value of x_m, x_r and computing the set of points in L can be done in $O(r)$ time and the other steps take constant time. So Find_Coordinate takes $O(r)$ time to



```
Procedure Find Coordinate $(j, b, d_d)$

    Let $d_\ell$ be the label of bag $b$
    Let $x_m$ be the smallest $j$th coordinate among all points in $b$
    Set $L$ to contain all points in $b$ with $j$ coordinate $x_m$
    Compute the second-minimal $j$ coordinate $x_r > x_m$
    Let $d_\Delta = x_m - x_r$
    For $z = 1, 2, 3$
        Let $\vec{v_z} = \vec{0}$ except set $v_j = \frac{(d_\ell+d_d)^2}{2d_\Delta} + d_\ell + d_d + z$
        $\ell_z = \mathtt{MI} - \mathtt{MQ}(b, x \cdot \vec{v}_j)$
        Let $y_z$ be the distance corresponding to label $\ell_z$
    Let $f$ be the parabola of form $y + (x - m)^2$ that contains the
        points $(v_0, y_0), (v_1, y_1), (v_2, y_2)$
    Return $m + x_m$
```

Fig. 4. The procedure Find_Coordinate searches for the coordinate of the target in dimension *j*. Recall that d_d is the maximum distance between any two points in *b*.



```
Algorithm Find Target $(b)$
    Find distances between all $r$ points of $b$ to compute the diameter $d_d$
    For each of the $n$ dimensions $k = 1, \ldots, n$
        Let $v_i$ = Find Coordinate($k$,$b$,$d_d$)
    Return $\vec{v} = (v_1, \ldots, v_n)$
```

Fig. 5. The algorithm Find_Target. Note that all bags created are linear transformations of the original bag *b* provided by the adversary.

compute the *j*th component v_j of $\vec{v}$. It takes $O(nr^2)$ time to compute d_d since there are $O(r^2)$ pairs of points and it takes $O(n)$ time to compute the distance between a pair of points. The loop in Find_Target takes $O(r)$ time for each of the *n* iterations. So the overall time complexity is $O(nr^2)$ with the dominant cost being that to compute d_d. We know from Lemma 4 that the distances y_0, y_1 and y_2 are determined by translations of the same point in the bag. So Lemma 3 proves that each element v_j of $\vec{v}$ is the coordinate of the target point in dimension *j*. □

5. Concluding remarks

In this paper, we present some hardness results and a positive result for learning in a real-valued MI learning model. We hope that this work will be the beginning of a theoretical study of learning in the real-valued MI model and eventually lead to improved algorithms for applications such as drug discovery. There are many interesting open problems. For example, are there nontrivial distributional assumptions, for which there is an efficient PAC learning (or on-line learning) algorithm to approximate the target point from real-valued MI data? Similarly, can hardness results be shown for more restricted distribution? Finally, are there alternate definitions for a MI membership query that better capture the physical constraints of the drug-discovery application.

Acknowledgments

Dr. Goldman received support from NSF Grants CCR-9988314 and IDM-0329241, and a Boeing-McDonnell Foundation grant. Dr. Dooly was supported in part by NSF Grant CCR-9988314. Dr. Kwek was supported by UTSA Faculty Research Award 14-7518-01.

References

[1] R.A. Amar, D.R. Dooly, S.A. Goldman, Q. Zhang, Multiple-instance learning of real-valued data, in: Proceedings of the Eighteenth International Conference on Machine Learning, vol. 18, San Francisco, CA, Morgan Kaufmann, Los Altos, CA, 2001, pp. 3–10.

[2] S. Andrews, I. Tsochantaridis, T. Hofmann, Support vector machines for multiple-instance learning, in: S. Becker, S. Thrun, K. Obermayer (Eds.), Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002, December 9–14, 2002, Vancouver, British Columbia, Canada], MIT Press, 2003.

[3] P. Auer, P.M. Long, A. Srinivasan, Approximating hyper-rectangles: learning and pseudo-random sets, in: Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing, vol. 29, ACM, New York, 1997, pp. 314–323.

[4] R.S. Berry, S.A. Rice, J. Ross, Physical Chemistry (Intermolecular Forces), Wiley, New York, 1980 (Chapter 10).

[5] A. Blum, A. Kalai, A note on learning from multiple-instance examples, Mach. Learning 30 (1998) 23–29.

[6] T.G. Dietterich, R.H. Lathrop, T. Lozano-Pérez, Solving the multiple-instance problem with axis-parallel rectangles, Artif. Intell. 89 (1–2) (1997) 31–71.

[7] S.A. Goldman, S.D. Scott, Multiple-instance learning of real-valued geometric patterns, Ann. Math. Artif. Intell., to appear.

[8] A.N. Jain, T.G. Dietterich, R.L. Lathrop, D. Chapman, R.E. Critchlow, B.E. Bauer, T.A. Webster, T. Lozano-Perez, Compass: a shape-based machine learning tool for drug design, J. Comput. Aided Molecular Design 8 (6) (1994) 635–652.

[9] M. Kearns, Efficient noise-tolerant learning from statistical queries, in: Proceedings of the Twenty-fifth Annual ACM Symposium on Theory of Computing, vol. 25, ACM Press, New York, NY, 1993, pp. 392–401.

[10] P.M. Long, L. Tan, PAC learning axis-aligned rectangles with respect to product distributions from multiple-instance examples, Mach. Learning 30 (1998) 7–21 (Earlier version in COLT96).

[11] O. Maron, Learning from ambiguity, AI technical report 1639, MIT, 1998.

[12] O. Maron, T. Lozano-Pérez, A framework for multiple-instance learning, Neural Inform. Process. Systems 10 (1998).

[13] O. Maron, A.L. Ratan, Multiple-instance learning for natural scene classification, in: Proceedings of the Fifteenth International Conference on Machine Learning, vol. 15, San Francisco, CA, Morgan Kaufmann, Los Altos, CA, 1998, pp. 341–349.

[14] S. Ray, D. Page, Multiple instance regression, in: Proceedings of the Eighteenth International Conference on Machine Learning, vol. 18, San Francisco, CA, 2001, Morgan Kaufmann, Los Altos, CA, 2001, pp. 425–432.

[15] G. Ruffo, Learning Single and Multiple Instance Decision Trees for Computer Security Applications, Ph.D. Thesis, Department of Computer Science, University of Turin, Torino, Italy, 2000.

[16] J. Wang, J.D. Zucker, Solving the multiple-instance problem: a lazy learning approach, in: Proceedings of the Seventeenth International Conference on Machine Learning, vol. 18, San Francisco, CA, 2000, Morgan Kaufmann, Los Altos, CA, pp. 1119–1125.

[17] M.K. Warmuth, J. Liao, G. Ratsch, M. Mathieson, S. Putta, C. Lemmen, Support vector machines for active learning in the drug discovery process, J. Chem. Inform. Sci. 43 (2) (2003) 667–673.

[18] Qi Zhang, S.A. Goldman, EM-DD: an improved multiple-instance learning technique, in: NIPS 2001, 2001, pp. 1073–1080.